Abstract—The problem of coverage of known space by a

mobile robot has many applications. Of particular interest is

providing a solution that guarantees the complete coverage

of the free space by traversing an optimal path, in terms

of the distance travelled. In this paper we introduce a new

algorithm based on the Boustrophedon cellular decomposition.

The presented algorithm encodes the areas (cells) to be covered

as edges of the Reeb graph. The optimal solution to the Chinese

Postman Problem (CPP) is used to calculate an Euler tour,

which guarantees complete coverage of the available free space

while minimizing the path of the robot. In addition, we extend

the classical solution of the CPP to account for the entry point

of the robot for cell coverage by changing the weights of the

Reeb graph edges. Proof of correctness is provided together

with experimental results in different environments.

The task of covering a bounded region of space is common

to many problems such as de-mining, vacuum cleaning, lawn

mowing and automated painting. One coverage application

that has proven to be extremely successful is the task of

robotic vacuum cleaning. The iRobot1 Roomba vacuum

cleaning robot uses a variety of different strategies, such

as random walk, wall-following, and the seed-spreader algorithm

[1], to achieve, probabilistically, coverage of the

whole floor space. The sale of more than two million robots

highlights the importance and impact of this application.

In the above described applications the problem of robotic

coverage of free space is defined as follows: the robot

has to pass an end effector, or a sensor, which in most

cases is the body of the mobile robot, over all available

free space. For example, during mine detection, the robot

has to ensure that every location that is not covered by

an obstacle is inspected and the position of the discovered

mines recorded. In such an application it is of paramount

importance to ensure completeness; no accessible area should

be left uncovered.

Depending on the target application, the proposed approaches

can be characterized according to different requirements.

One important division is between algorithms

that require a map of the environment, which describes the

occupied and free space, and the algorithms that are capable

of covering an unknown environment. For unknown environments

it is impossible to provide a criterion of optimality as

for any design choice a counter-example environment could

be constructed. Nevertheless, during coverage of unknown

environments it is important in terms of efficiency to avoid

repeated coverage [2] as much as possible. For known

environments it is important to accomplish the coverage

task in minimum time. A subset of all coverage planning

schemas are said to be complete, meaning they guarantee

the entire free space will be covered by the provided plan.

Such schemas are often based on rigorous representations

of the free space such as exact cellular decompositions or

spanning trees on which coverage planning is simplified.

In this paper we present a new algorithm for the complete

coverage of an arbitrary known environment. Our algorithm

is using the Boustrophedon cellular decomposition to divide

the free space in cells; then, the solution to the Chinese

Postman Problem is used to calculate the order in which

the cells are going to be covered. We extend the classical

Boustrophedon decomposition by splitting some cells in half

along the direction of coverage. As a result the robot is

capable of covering each cell, or sub-cell, in such order

that no cell is traversed twice, and at the end the robot has

returned near the starting position. Further improvement in

performance is achieved by calculating the width of each

cell and estimating the position of the robot at the end

of the coverage cycle. As in most previous treatments of

coverage, the robot operates under the assumption of accurate

localization. Experimental results in different environments

illustrate the efficiency of our approach.

Related work would be discussed in the next section,

including a brief overview of the Boustrophedon cellular

decomposition that provides the basic component to our

approach, as well as related concepts from graph theory.

In Section III a detailed description of our algorithm is

presented, together with a sketch of the optimality proof.

Section IV demonstrates the experimental verification of

our approach in different classes of simulated environments.

Section V contains conclusions and directions of future work.

II. BACKGROUND

Choset and Pignon first introduced a rigorous extension to

the Seed Spreader algorithm [1] under the name of Boustrophedon

Decomposition [3]. The work was further developed

by Acar et al. [4] with experimental verification and for

a variety of control Morse functions. The Boustrophedon

family of algorithms guarantees the complete coverage of an

unknown environment with no claims on the distance travelled.

Section II-A provides an outline of these algorithms

as they form one of the building blocks of our approach.

Butler et al. [5] achieved complete coverage of unknown

rectilinear environments using a square robot with contact

sensing. They performed an on-line decomposition where

each cell, in the shape of a rectangle, was formed such that

it could be covered completely by back-and-forth motions

performed parallel to one of the walls of the environment.

A different approach was followed by Huang [6]: the

environment was subdivided in different regions aiming to

minimize the amount of rotations the robot has to perform.

The total path travelled was ignored. Yao proposed an improved

algorithm [7] that reduces the path travelled compared

to [6]. Illustrative examples were shown, but no proof of

optimality. More recently, Kang et al. [8] proposed a scheme

where a set of precalculated motion strategies are selected

in order to minimize repeat coverage. In all of the above the

claims are only supported by examples in simulation.

Gabriely and Rimon [9], [10] used a grid based approach

for planning a complete coverage path. The main requirement

is for the environment to be decomposable to a grid. For

known terrain, algorithms proposed by Zheng et al. guarantee

a performance of at most eight times the optimal cost [11].

This is achieved by superimposing a grid with each grid

cell having the size of four footprints of the robot. Our

approach eliminates the grid restrictions on the environment

and guarantees an optimal path.

Several researchers have worked on the problem of distributing

a group of nodes (mobile sensors) such that they

achieve the maximum coverage of an area of interest. The

robots, after reaching their position, do not move unless there

are dynamic changes in the environment. In particular, Cort´es

et al. proposed an approach which utilizes the centers of

the Voronoi cells [12], [13]. Using artificial potential fields,

Howard et al. control the robots to move away from each

other, thus increasing the covered area [14]. More recently,

Schwager et al. proposed a unifying scheme for multi-robot

coverage which combines the previous methods [15]. All

of the above are concerned with the sensor coverage at the

final position of the robots e.g. in surveillance operations

as opposed to the path-planning problem more common to

applications such as de-mining, vacuum cleaning, etc.

Other approaches used genetic algorithms [16] and landmarks

[17] to improve the speed of coverage. Easton and

Burdick [18] used a variant of the Chinese Postman Problem

to solve the problem of boundary coverage. For information

on several more algorithms on coverage see [19]. In addition,

many authors have worked on the area of multi-robot

coverage; for an extensive survey please refer to [20].

A. Boustrophedon Cellular Decomposition

Our approach utilizes the concepts of Boustrophedon

Cellular Decomposition of unknown environments to optimally

schedule the order of coverage. To better describe our

algorithm, the following terms from single robot coverage

are used: slice, cell, sweep direction, and critical point [21],

[4]; see Fig. 1a. The Boustrophedon decomposition [3] is a

type of Morse decomposition where the slice is a line. The

robot follows the intersection of the slice and the area to be

covered, thus covering the area with vertical back and forth

motions. A cell is a region defined by the Boustrophedon

decomposition where slice connectivity does not change. In

other words, no obstacle breaks the connectivity of the slice

inside each cell. Sweep direction refers to the direction in

which the slice is swept. Lastly, a critical point represents

a point on an obstacle which causes a change in the slice

connectivity. Thus, the free space is divided into regions

(cells) of constant slice connectivity, each of which can be

covered with a vertical back and forth motion.

Another concept used here is the Reeb graph [21], [4]. A

Reeb graph is a graph representation of the target environment

where the nodes represent the critical points and the

edges represent the cells; see Fig. 1b. Due to the nature of

the Boustrophedon decomposition, all concave critical points

are connected to exactly one cell, i.e., a node of degree one

in the Reeb graph. Similarly, all convex critical points are

connected to exactly three cells, i.e. a node of degree three

in the Reeb graph.

B. Graph Theory

Different algorithms from graph theory have been used in

robotics to guide exploration [22], mapping [23], [24], and

coverage [25] in the past. Edmonds and Johnson [26] present

an overview of different graph algorithms that are directly

applicable to the problem of optimal coverage of a known

environment.

An Euler tour is a circuit that covers every edge in a

graph exactly once. Euler demonstrated that a necessary and

sufficient condition for the existence of such a tour in a

graph is that all nodes of the graph have even degree – such

graphs are called Eulerian Graphs. A similar problem is the

Chinese Postman Problem (CPP): find a shortest tour that

traverses every edge at least once. If a graph is Eulerian

then all of its Euler tours are solutions to the CPP. For non-

Eulerian graphs, a standard approach to solving the CPP is to

double selected edges in the graph – i.e. given an edge e that

connects nodes Vi and Vj, add an edge f that also connects

nodes Vi and Vj – to make the resulting graph Eulerian

and to then choose one of the Euler tours as the solution.

Different strategies can be applied to determine which edges

to double. All of the new edges will by definition be part

of the Euler tour; hence, the challenge is to choose edges

such that the total cost – the sum of the individual costs

of all the edges – of this Euler tour be minimized. The

constraints and objective function of a linear programming

system that can be solved to choose the edges to double is

described in [26]. In addition, these edges can be chosen

by utilizing algorithms from matching theory. The simplest

of these algorithms essentially determines the shortest paths

between every odd-node (i.e. nodes of odd degree) in the

original graph, the total costs of which are used to determine

which edges should be doubled. Consequently, an Euler tour

can be extracted from an Eulerian graph. A simple and

wildly inefficient algorithm was proposed [26]: let G be the

original graph. Repeatedly choose and remove edges whose

deletion would not disconnect G unless there is no other

choice until G is empty. The sequence of chosen edges

is the Euler tour. More efficient, but also more complex,

algorithms iteratively build disjoint non-Euler tours, which

they connect in a certain fashion. When there are no more

unvisited edges, the constructed sequence is an Euler tour.

For more information please refer to [26].

Next, the optimal coverage algorithm of an arbitrary

known environment is presented.

III. OPTIMAL COVERAGE ALGORITHM

The optimal coverage algorithm is divided in two parts:

first, an off-line analysis of the environment, construction

of the Boustrophedon Cellular Decomposition (BCD) and

the Reeb Graph (RB), formulating and solving of a linear

programing problem for the construction of an Euler tour;

and second, on-line coverage that uses the sequence of edges

in the Euler tour to guide the coverage from one cell to the

next. Each component will be discussed next

A. Construction of the Reeb Graph

The input to our algorithm is a bitmap representation of

the environment. This choice comes naturally as the testing

environment that we employ, Player/Stage, uses the same

representation. In addition, a bitmap representation provides

maximum flexibility in the modeling of the free/occupied

space, without any restrictions on the shape of the obstacles.

As can be seen in the experimental results section, different

environments, such as indoor office space, outdoor, sparsely

populated areas, as well as completely unstructured areas

were used to verify the performance of our approach.

A common assumption in most cellular decomposition

algorithms is that no two critical points change the slice

connectivity at the same time. This assumption is trivial to

enforce by a small rotation of the sweep direction. It is worth

noting that in this work we extend the notion of a critical

point to include critical regions; for example a whole wall

that changes the slice connectivity. See for example the left

and right walls in Fig. 3a. In a structured environment, such

as an office space, it is beneficial to perform coverage parallel

to the walls. The above mentioned extension allows it.

The first task is to sweep the bitmap. Without loss of

generality we assume a sweep direction along the x-axis and

record all the critical points/regions. During the sweep, the

location of the cells is also recorded. Finally, the critical

points and the cells are encoded in the Reeb Graph. The

resulting Reeb Graph G =< V;E > is used as input to the

next step of the algorithm that calculates an Euler tour.

B. Construction of the Euler Tour

The primary contribution of our algorithm is using the

solution to the CPP in order to find the optimal order, in

terms of distance travelled, in which the cells are covered.

Given the Reeb graph, the next task would be to calculate

an Euler tour. As mentioned in Sect. II-B, this can be

achieved by doubling selected edges of the Reeb Graph.

Consequently, when the robot covers one of these doubled

edges, it will split the cell into top and bottom sub-cells

and assign each sub-cell to one of the two doubled edges.

From [26], no edge would be duplicated more than once.

In addition, we bias the edge duplication in cells that are

wider in order to facilitate easier coverage2. An instance of

a linear programming problem described in Eq. 1 is created.

The solution to the LP represents the Euler tour that is the

routing of the robot.

xe is integer e 2 E; wn is integer n 2 V;

xe 0; e 2 E; wn 0; n 2 V;

å

e2E

anexe􀀀2wn = bn; n 2 V;

z = å

e2E

cexe is minimized. (1)

where: xe is the number of added copies of edge e in the

solution. åe2E anexe represents the number of added edges to

node n 2 V in the solution. Note that for the solution to be

an Eulerian graph, an odd number of edges has to be added

to nodes with odd degree and an even number of edges has

to be added to nodes with even degree. bn is 1 for nodes

with odd degree and 0 for nodes with even degree. wn is a

variable that will force åe2E anexe to be odd for odd nodes

and even for even nodes. ane is 1 if node n meets edge e,

and 0 otherwise; ce is the cost of edge e.

The GLPK3 library was used to solve the above linear

programing problem. The solution is used to guide the robot

from cell to cell, and each cell is covered using a modified

version of the Boustrophedon Coverage.

C. Boustrophedon Coverage

The simple back-and-forth motion used for covering the

interior of a cell is well documented in the literature [21], [4],

[3], [19], and covering it is beyond the scope of this paper.

The novel contribution of our algorithm is the treatment of

the cells, which had their corresponding Reeb Graph edge

doubled in order to generate an Euler tour. These cells are

necessary to be traversed twice. The coverage algorithm is

modified to cover during the first traversal the top (or bottom)

half of the cell, and in the second traversal to cover the

bottom (or top) half of the cell correspondingly, see Fig.

2. The choice of which half to cover is dictated by the

position of the robot at the end of the previous coverage

task. Therefore, at the end of the coverage, the robot will

have covered the complete cell without duplicating any work.

It is worth noting that the robot is operating in a known

environment. As such, when the robot enters a cell to cover

it, it has the information on the length and the height of that

cell. Consequently, an informed decision can be made, if the

robot covers only the top half of the cell, where to stop each

downward motion; see Fig. 2. More formally, for a cell ci

with width wi and a slice width of ws there are going to be

k = wi=ws vertical motions to cover the ci. When the robot

begins a downward motion, it will cover only hj distance,

see Eq. 2.

hj = hi

j

k􀀀 j􀀀1

k

; j = 0: : :k􀀀1 (2)

where, hi

j, is the length of the slice at vertical motion j,

which is calculated by the height of cell ci at that point.

By varying the height of the coverage progressively, the

robot is capable of controlling the exit point from each cell,

thus minimizing the distance to the entry point to the next

cell.

D. Proof of Correctness

The correctness of the algorithm and the optimality follow

directly from the properties of the cellular decomposition and

the Euler tour used. By definition, the Reeb graph provides

a complete model of the environment. By ensuring that each

edge of the graph is traversed (covered) we guarantee that

all available free space has been covered. The Euler tour

resulting from the doubling of selected edges provides an

order in which the cell of the Reeb graph should be visited.

By definition, there is no edge of the Euler tour that is

traversed twice, which means that no area is covered twice.

Therefore, the proposed algorithm is optimal, as all free

space is covered exactly once.

During coverage the border areas adjacent to obstacles

could be wider than a single pass but narrower than two.

In such cases the robot sensor would cover these areas

twice. In general, these areas, with the exception of “fractallike”

counterexamples, are an order of magnitude smaller

compared to the interior area to be covered. Therefore, repeat

coverage of the boundaries does not affect the optimality.

It is worth noting that the structure of the Reeb graph

is exploited in order to provide a polynomial time solution

to the complete coverage, minimum distance, path planning

problem.

IV. EXPERIMENTAL RESULTS

Numerous experiments were conducted for a variety

of environments using the robotic simulation package

Player/Stage4. A simulated Pioneer robot was used to perform

coverage of all the available free space. In particular,

three different classes of environments were used as test

cases. First, office like environments such as the one in Fig.

3a were used, inspired by the service robotics applications.

Second, open fields with sparse tree and rock like obstacles

were used, the target application being humanitarian demining;

see Fig. 3d. Finally, arbitrary environments with convex

and concave objects were used to ensure the performance

of the algorithm under arbitrary conditions; see Fig. 3g.

Figure 3a illustrates the application of our algorithm in an

indoor environment, such as an office building. The solution

presented in Fig. 3b guides the robot to cover each one of

the thirty eight cells and sub-cells, and return at the starting

position. Figure 3c shows the path of the robot as it covers

sub-cell 1 and cell 2. A similar structure is followed in the

rest of Fig. 3. Figure 3d shows a sparsely populated area.

The free space is subdivided in fifty four different cells;

Figure 3f shows the path of the robot while covering areas

1 to 5. Finally, Fig. 3g-i demonstrates the application of our

algorithm in an environment populated by arbitrary concave

and convex obstacles, where the resulting Euler tour results

in twenty one cells and sub-cells.

Figure 4 presents another example of an arbitrary environment.

Figure 4a shows the Boustrophedon Cellular

Decomposition as dashed lines, together with the resulting

Euler tour. Figure 4(b-f) presents a sequence of screen-shots

of the Stage simulation environment with the robot together

with a limited trace. In Fig. 4(b) the robot is covering the

cell 11. In Fig. 4(c) the top sub-cell (13) is being covered.

The robot continues to the top sub-cell (16), see Fig. 4(d).

The cell 18 and then the cell 20 are covered in Fig. 4(e) and

(f) accordingly.

V. CONCLUSIONS

In this paper we presented a new algorithm for the

complete coverage of a known arbitrary environment. The

algorithm guides a mobile robot through a sequence of

areas to be covered without wasting energy and time by

moving through already covered areas. The solution to the

Chinese Postman Problem from graph theory is adapted for

the calculation of the cell ordering. The single cell coverage

used in the Boustrophedon Cellular Decomposition algorithm

is modified in order to eliminate repeat coverage by splitting

selected cells into two components.

Experiments using real hardware are scheduled for the

near future in order to further validate our approach. A

iRobot Create mobile robot will be used in combination with

an overhead camera for accurate localization. Furthermore, we are planning to investigate the effect of rotations versus

the distance travelled in order to propose time optimal

strategies extending the current approach.

Automated coverage by mobile robots is an area with

many applications. With the expected increase in service

robots in private households as well as in public spaces,

improving coverage performance is essential. The presented

algorithm is a significant step in this direction.

WeakNess

1. **Real-world Testing**: While the paper includes simulations to validate the algorithm, real-world experiments with physical robots would provide a more comprehensive understanding of the algorithm's effectiveness in practical scenarios.
2. **Complexity and Scalability**: The paper could explore the algorithm's complexity and scalability in more detail. Understanding how well the algorithm scales with larger and more complex environments would be beneficial, especially for practical applications.
3. **Robustness to Environmental Changes**: The algorithm is designed for known environments. Extending the algorithm to adapt to dynamic changes in the environment would make it more versatile and practical for real-world applications where environments are not static.
4. **Energy Efficiency**: The paper focuses on optimizing the path length, but another important aspect in mobile robotics is energy efficiency. Future work could include energy considerations, finding a balance between the shortest path and energy-efficient movements, especially for battery-powered robots.
5. **Integration with Other Robotic Systems**: Exploring how this algorithm could be integrated with other robotic systems, such as obstacle avoidance or interactive systems, would increase its applicability in complex environments.
6. **Comparison with Other Methods**: A more detailed comparison with other existing coverage algorithms would help in understanding the relative advantages and potential drawbacks of this approach.
7. **Experimentation with Real Hardware**: The paper acknowledges the need for further validation of the approach using real hardware. The authors mention plans for experiments using an iRobot Create mobile robot in combination with an overhead camera for accurate localization. This suggests that, while the algorithm has been tested in simulations, real-world testing and refinement are necessary.
8. **Investigation of Rotations Versus Distance Travelled**: The authors plan to investigate the effect of rotations versus the distance traveled to propose time-optimal strategies. This indicates an area for enhancing the efficiency of the coverage algorithm, especially in terms of optimizing the time required for coverage tasks.
9. **Generalization of the Approach**: While the paper presents a significant step in coverage algorithms for mobile robots, there is an implied need for further development to generalize the approach for a wider range of applications and environments. This includes adapting the algorithm to various types of robotic platforms and environmental conditions.
10. **Optimization in Specific Scenarios**: The document discusses the algorithm's performance in specific scenarios, like office spaces or sparsely populated areas, but further optimization might be required for more complex or dynamic environments.
11. **Handling of Boundary Areas**: The paper mentions that during coverage, border areas adjacent to obstacles might be wider than a single pass but narrower than two, potentially leading to repeated coverage in these areas. While this is stated not to affect the overall optimality, refining the approach to handle such boundary conditions more effectively could be an area for improvement.

**Algorithm Complexity**: While the paper discusses the algorithm's theoretical basis, the computational complexity and practical feasibility of implementing it on robots with limited processing capabilities might be challenging.